

1        **Amendment to the Claims**

2        **In the Claims:**

3        Please amend Claim 21 as follows:

4        1. (Original) A method for padding a macroblock of a video object, comprising the steps of:

5                (a) horizontally scanning shape data for successive rows of the macroblock with a  
6 host processor to detect any transparent pixels and any opaque pixels contained in each row;

7                (b) at each occurrence of one or more transparent pixels in a row having at least  
8 one opaque pixel, detected while horizontally scanning the shape data, determining a horizontal  
9 primitive with the host processor, as a function of a number of successive transparent pixels detected  
10 in the row;

11                (c) communicating the horizontal primitive to a coprocessor for horizontal  
12 padding of the row of the macroblock of step (b), as a function of texture data associated with one or  
13 more opaque pixels that are adjacent to said one or more transparent pixels, producing horizontally  
14 padded pixels;

15                (d) for each row that includes only transparent pixels, determining a vertical  
16 primitive as a function of a number of adjacent rows that include only transparent pixels; and

17                (e) communicating the vertical primitive to the coprocessor for vertical padding of  
18 the macroblock as a function of the texture data associated with pixels disposed in one or more  
19 adjacent rows, producing a fully padded macroblock.

20        2. (Original) The method of Claim 1, wherein the step of determining the horizontal  
21 primitive, comprises the step of determining a coordinate of a first opaque pixel that is adjacent to the  
22 one or more transparent pixels in the horizontal row of the macroblock, if the horizontal row includes  
23 at least one opaque pixel.

24        3. (Original) The method of Claim 2, further comprising the step of determining a coordinate  
25 of a second opaque pixel that is adjacent to the one or more transparent pixels in the horizontal row of  
26 the macroblock if the sequential transparent pixels are bounded by opaque pixels on two sides in the  
27 horizontal row.

28        4. (Original) The method of Claim 2, further comprising the step of updating the coordinate  
29 of the first opaque pixel if more than one opaque pixel occurs sequentially within the horizontal row.

30        //

1       5. (Original) The method of Claim 1, wherein the step of determining the horizontal  
2 primitive comprises the steps of:

3               (a) flagging a row number of any horizontal row of the macroblock having only  
4 transparent pixels;

5               (b) assigning a dot primitive as the horizontal primitive for padding any  
6 transparent pixel that is not adjacent to any other transparent pixel; and

7               (c) assigning a horizontal line as the horizontal primitive if the number of  
8 successive transparent pixels is greater than one.

9       6. (Original) The method of Claim 1, wherein the step of determining the vertical primitive,  
10 comprises the steps of:

11               (a) determining a number of successive flagged rows in the macroblock, wherein a  
12 flagged row is identified during the step of horizontally scanning and is a horizontal row having only  
13 transparent pixels; and

14               (b) if the macroblock includes at least one flagged row, determining a row  
15 coordinate of a first row that comprises at least one opaque pixel and that is adjacent to said at least  
16 one flagged row.

17       7. (Original) The method of Claim 6, further comprising the step of determining a coordinate  
18 of a second row that is adjacent to the successive flagged rows in the macroblock if the successive  
19 flagged rows are between rows each having at least one opaque pixel.

20       8. (Original) The method of Claim 6, wherein the step of determining the row coordinate of  
21 the first row comprises the steps of identifying any successive rows that comprise at least one opaque  
22 pixel until the row having at least one opaque pixel and which is adjacent to said at least one flagged  
23 row is identified.

24       9. (Original) The method of Claim 5, wherein the step of determining a vertical primitive  
25 comprises the steps of:

26               (a) determining that extended padding must be used for the macroblock if all rows  
27 of the macroblock are flagged as having only transparent pixels; and if not,

28               (b) assigning a line-v primitive as the vertical primitive for use in padding a  
29 flagged row if the flagged row is not adjacent to another flagged row; and

30       ///

(c) assigning a rectangle primitive as the vertical primitive if the number of successive flagged rows is greater than one.

10. (Original) The method of Claim 1, wherein the step of communicating the vertical primitive to the coprocessor, comprises the steps of:

(a) determining whether a latency period has been surpassed, said latency period being selected to be sufficiently long to enable the coprocessor to complete horizontal padding of a predefined number of macroblocks; and

(b) if the latency period has been surpassed, communicating the vertical primitive to the coprocessor along with at least one argument associated with the vertical primitive.

11. (Original) The method of Claim 1, further comprising the steps of:

(a) causing the coprocessor to horizontally pad any transparent pixels with texture data associated with a coordinate of one or more opaque pixels identified in the horizontal primitive communicated to the coprocessor; and

(b) causing the coprocessor to vertically pad transparent pixels with texture data associated with a coordinate of one or more opaque pixels identified in the vertical primitive communicated to the coprocessor.

12. (Original) The method of Claim 11, wherein the step of causing the coprocessor to horizontally pad comprises the steps of:

(a) causing the coprocessor to pad any transparent pixels identified in the horizontal primitive with texture data associated with a coordinate of one opaque pixel identified in the horizontal primitive if only one opaque pixel is identified in the horizontal primitive; and

(b) causing the coprocessor to pad any transparent pixels identified in the horizontal primitive with an average of texture data associated with coordinates of two opaque pixels identified in the horizontal primitive if two opaque pixels are identified in the horizontal primitive.

13. (Original) The method of Claim 11, wherein the step of causing the coprocessor to vertically pad comprises the steps of:

(a) causing the coprocessor to pad transparent pixels identified in the vertical primitive with texture data associated with a coordinate of one adjacent row identified in the vertical primitive if only one adjacent row is identified in the vertical primitive; and

111

(b) causing the coprocessor to pad any transparent pixels identified in the vertical primitive with an average of texture data associated with coordinates of two adjacent rows identified in the vertical primitive if two adjacent rows are identified in the vertical primitive.

14. (Original) The method of Claim 1, wherein the horizontal and vertical primitives are determined by the host processor utilizing a host memory in which the shape data are stored and the texture data are stored in a video memory that is utilized by the coprocessor for padding the macroblock.

15. (Original) A machine-readable medium storing machine instructions that cause a processor to perform the steps of Claim 1.

16. (Original) A system for padding a boundary macroblock of a video object, comprising:

- (a) a graphics coprocessor;
- (b) a host processor in communication with the graphics processor; and
- (c) a host memory in communication with the host processor, said host memory data defining the video object, a boundary portion of which is associated with the and said host memory storing machine instructions that cause the host processor to:

- (i) horizontally scan shape data for successive rows of the macroblock with the host processor to detect any transparent pixels and any opaque pixels contained in each row;

(ii) at each occurrence of one or more transparent pixels in a row having at least one opaque pixel, detected while horizontally scanning the shape data, determine a horizontal primitive with the host processor, as a function of a number of successive transparent pixels detected in the row;

(iii) communicate the horizontal primitive to the graphics coprocessor for horizontal padding of the row of the macroblock, as a function of texture data associated with one or more opaque pixels that are adjacent to said one or more transparent pixels, to produce horizontally padded pixels;

(iv) for each row that includes only transparent pixels, determine a vertical primitive as a function of a number of adjacent rows that include only transparent pixels; and

- (v) communicate the vertical primitive to the coprocessor for vertical padding of the macroblock as a function of the texture data associated with pixels disposed in one or more adjacent rows, to produce a fully padded macroblock.

1       17. (Original) The system of Claim 16, further comprising a video memory in  
2 communication with the graphics coprocessor, said video memory storing texture data used to pad the  
3 macroblock.

4       18. (Original) The system of Claim 16, further comprising a buffer in communication with  
5 the host processor and with the graphics coprocessor, said buffer temporarily storing vertical  
6 primitives determined by the host processor until a predefined latency period is surpassed, after  
7 which the vertical primitives are communicated to the graphics coprocessor.

8       19. (Original) The system of Claim 16, further comprising a data bus providing  
9 communication from the host processor to the graphics coprocessor, said data bus conveying each  
10 horizontal primitive and each vertical primitive from the host processor to the graphics coprocessor.

11       20. (Original) The system of Claim 19, wherein the data bus is one of an accelerated  
12 graphics port (AGP) bus and a peripheral component interconnect (PCI) bus.

13       21. (Currently Amended) A method for padding a macroblock of a video object, comprising  
14 the steps of:

15               (a) determining a graphics primitive with a host processor based on shape data  
16 representing the video object that are accessible by the host processor, said shape data being stored in  
17 a host memory that is accessible by the host processor;

18               (b) communicating the graphics primitive to a coprocessor via a data bus that  
19 couples the host processor in communication with the coprocessor; and

20               (c) padding the macroblock with the coprocessor, based on the graphics primitive  
21 and on texture data that are accessible by the coprocessor.

22       22. (Original) The method of Claim 21, wherein the shape data indicates:

23               (a) any transparent pixels in the macroblock; and  
24               (b) any opaque pixels in the macroblock.

25       23. (Original) The method of Claim 22, wherein the step of determining the graphics  
26 primitive comprises the steps of:

27               (a) determining a number of one or more transparent pixels to be padded from the  
28 shape data;

29               (b) determining coordinates of at least one transparent pixel included in the one or  
30 more transparent pixels;

(c) determining coordinates of at least one opaque pixel having texture data that will be used for padding said one or more transparent pixels;

(d) selecting a primitive that encompasses the one or more transparent pixels that were determined; and

(e) communicating the primitive, the coordinates of the at least one transparent pixel, and the coordinates of the at least one opaque pixel to the coprocessor.

24. (Original) The method of Claim 23, wherein the step of padding the macroblock, comprises the steps of:

(a) obtaining texture data corresponding to the coordinates of the at least one opaque pixel;

(b) determining a padding texture value for each of the one or more transparent pixels from the texture data corresponding to the coordinates of the at least one opaque pixel; and

(c) rendering the selected primitive to pad each of said one or more transparent pixels of the macroblock.

25. (Original) The method of Claim 21, wherein the step of padding the macroblock accelerates Motion Picture Experts Group level 4 (MPEG-4) video decoding.

26. (Original) The method of Claim 21, wherein the coprocessor is capable of performing MPEG-2 video decoding.

27. (Original) A machine-readable medium storing machine instructions that cause a processor to perform the steps of Claim 21.

28. (Original) A system for padding a macroblock of a video object, comprising:

(a) a host processor;

(b) a host memory in communication with the host processor, said host memory

(i) shape data for the video object; and

(ii) machine instructions that cause the host processor to determine a  
and at least one argument for the graphics primitive based on the shape data;

(c) a coprocessor in communication with the host processor to receive the graphics primitive and the at least one argument; and

111

1 (d) a graphics memory in communication with the coprocessor, said graphics  
2 memory storing:

3 (i) texture data defining a texture of the video object; and  
4 (ii) machine instructions that cause the coprocessor to pad the macroblock  
5 based on the graphics primitive, the at least one argument, and the texture data.

6 29. (Original) The system of Claim 28, further comprising a data bus in communication with  
7 the host processor and the coprocessor, said data bus carrying the graphics primitive and the at least  
8 one argument from the host processor to the coprocessor.

9       30. (Original) The system of Claim 29, wherein the data bus is one of an accelerated  
10      graphics port (AGP) bus and a peripheral component interconnect (PCI) bus.

11        31. (Original) The system of Claim 28, further comprising a buffer in communication with  
12 the host processor and with the coprocessor, said buffer temporarily storing the graphics primitive  
13 determined by the host processor until a predefined latency period is surpassed, after which the  
14 graphics primitive is communicated to the coprocessor.

32. (Original) The system of Claim 28, wherein the shape data indicates:

16 (a) any transparent pixels in the macroblock; and  
17 (b) any opaque pixels in the macroblock.

18 33. (Original) The system of Claim 32, wherein the machine instructions that cause the host  
19 processor to determining the graphics primitive cause the host processor to:

20 (a) determine a number of one or more transparent pixels to be padded from the  
21 shape data;

22 (b) determine coordinates of at least one transparent pixel included in the one or  
23 more transparent pixels;

24 (c) determine coordinates of at least one opaque pixel having texture data that will  
25 be used for padding said one or more transparent pixels;

26 (d) select a primitive that encompasses the one or more transparent pixels that  
27 were determined; and

28 (e) communicate the primitive, the coordinates of the at least one transparent pixel, and  
29 the coordinates of the at least one opaque pixel to the coprocessor.

30 | //

1           34. (Original) The system of Claim 33, wherein the machine instructions that cause the  
2 coprocessor to pad the macroblock cause the coprocessor to:

3           (a)    obtain texture data corresponding to the coordinates of the at least one opaque  
4 pixel;  
5           (b)    determine a padding texture value for each of the one or more transparent  
6 pixels from the texture data corresponding to the coordinates of the at least one opaque pixel; and  
7           (c)    render the selected primitive to pad each of said one or more transparent pixels  
8 of the macroblock.

9           35. (Original) The system of Claim 28, wherein the machine instructions that cause the  
10 coprocessor to pad the macroblock cause the coprocessor to accelerate MPEG-4 video decoding.

11           36. (Original) The system of Claim 28, wherein the coprocessor is capable of performing  
12 MPEG-2 video decoding.

13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30